PAT-NO: JP401302288A

DOCUMENT-IDENTIFIER: JP 01302288 A

TITLE: POWER-RESIDUE CALCULATION SYSTEM

PUBN-DATE: December 6, 1989

INVENTOR-INFORMATION:
NAME
KAWAMURA, SHINICHI
SHINPO, ATSUSHI

ASSIGNEE-INFORMATION:

NAME	COUNTRY
TOSHIBA CORP	N/A

APPL-NO: JP63050901

APPL-DATE: March 4, 1988

INT-CL (IPC): G09C001/00, G06F007/72

US-CL-CURRENT: 708/490

ABSTRACT:

PURPOSE: To speed up arithmetic processing by using a multiplier and a residue calculator for cumulation processing simultaneously.

CONSTITUTION: When an integer given by an equation II is calculated from an integer given by an equation I, the number N of a modulus is passed from a controller 21 to a residue computing element 15. Then, 1 is stored as an initial value in a C register 11 and M is stored in a W register 13. When the least significant digit bit e<SB>0</SB> of a power number E

is '0', RmodN is
found by using the residue calculator 15 and stored in registers 12 and 13 and
then the result is multiplied by using a multiplier 14, so that the result is
stored in the register 12. When the e<SB>0</SB> is '1', the cumulative
processing is carried out simultaneously with said squaring processing and the
cumulation processing is performed by using the multiplier 14 first then the
residue calculator 15 to find CmodN. Thus, the multiplier and residue
calculator 15 are used for different processes at the same time, thereby
speeding up the processes.

⑲日本国特許庁(JP)　⑪特許出願公開

⑫公開特許公報(A)　平1-302288

⑤Int.Cl.⁴	識別記号	庁内整理番号	㊸公開　平成1年(1989)12月6日
G 09 C 1/00		7368-5B	
G 06 F 7/72		7056-5B	

審査請求　未請求　請求項の数 1　(全5頁)

㊹発明の名称　べき乗剰余計算方式

㉑特　願　昭63-50901
㉒出　願　昭63(1988)3月4日

㉗発明者　川村　信一　神奈川県川崎市幸区小向東芝町1番地　株式会社東芝総合研究所内
㉗発明者　新保　淳　神奈川県川崎市幸区小向東芝町1番地　株式会社東芝総合研究所内
㉑出願人　株式会社東芝　神奈川県川崎市幸区堀川町72番地
㉔代理人　弁理士　鈴江　武彦　外2名

明　細　書

1．発明の名称

べき乗剰余計算方式

2．特許請求の範囲

第1、第2及び第3のレジスタ、乗算器、剰余計算器並びに制御手段を備え、与えられた整数M，N，E（$-\sum_{i=0}^{k} e_i \cdot 2^i$，$e_i$ー1又は0）から、C＝$M^E$ mod Nなるべき整数Cを計算するべき乗剰余計算方式において、

前記制御手段は、初期値として前記第1のレジスタに1、前記第2のレジスタにM^2、前記第3のレジスタにMをそれぞれ設定し、iを0から順に1ずつ増す度に、前記第2のレジスタの格納値と前記整数Nとを前記剰余計算器に供給しその法Nの演算結果を前記第2及び第3のレジスタに格納する第1のステップと、この第1のステップで求められた演算結果を前記乗算器に供給してその二乗値を求め前記第2のレジスタに格納する第2のステップとからなる二乗処理の実行を制御するとともに、

iを0から順に1ずつ増す度に、e_iが1であるときのみ、前記第1のステップと同時に、前記第1のレジスタの格納値と前記第3のレジスタの格納値とを前記乗算器に供給しその結果を前記第1のレジスタに格納するとともに、前記第2のステップと同時に、前記第1のレジスタの格納値と前記整数Nとを前記剰余計算器に供給しその法Nの演算結果を前記第1のレジスタに格納する累積処理の実行を制御するものであることを特徴とするべき乗剰余計算方式。

3．発明の詳細な説明

（産業上の利用分野）

この発明は、例えばRSA暗号化アルゴリズムに使用され、ビット数を増やすことなくべき乗剰余計算を実行するべき乗剰余計算方式に関し、特に演算処理の高速化を図れるようにしたべき乗剰余計算方式に関する。

（従来の技術）

RSA暗号化方式は、現在最も有望視されている公開鍵暗号方式である。このRSAのアルゴ

リズムは、C，M，N，Eを整数、Mを平文、Cを暗号文としたとき、

$$C = M^E \bmod N$$

によって表わすことができる。

ところで、上記アルゴリズムにおいて、C，M，Nは、通常150桁程度の大きな数であることから、上記アルゴリズムを実際にインプリメントする場合、M^Eを計算した後に剰余計算を行うのは現実的でない。そこで、べき指数Eを2進数（e_k，…，e_1，e_0）に展開し、その最下位ビットから最上位ビットまで、A×B mod Nの計算を繰返す手法が採用される。この手法の詳細を第4図に示す。まず、M，N，Eが与えられ、初期値としてCレジスタに1、RレジスタにMが与えられる（1）。そして、iを0からkまで1ずつ順次増やしながら（2，3）、$e_i = 1$であれば（4）、

$$C \leftarrow C * R \bmod N$$

なる累積処理を実行した後（5）、

$$R \leftarrow R * R \bmod N$$

なる二乗処理を実行し（6）、$e_i = 0$であれば（4）、上記二乗処理のみを実行する（5）。

このアルゴリズによれば、1つの処理サイクル内の累積処理（5）及び二乗処理（6）において、剰余計算を行って桁数を削減しながら、Cレジスタにべき乗剰余演算結果、Rレジスタに二乗演算結果をそれぞれ格納して処理が進められ、最終的にCレジスタ内に結果Cを求めることができる。

このアルゴリズムを実行する従来のべき乗剰余計算方式は、累積処理及び二乗処理を実行するため、2数の乗算を行う乗算器と、その乗算結果に対する剰余算を行う剰余計算器とを備えて構成される。そして、乗算は剰余算に先だって行われるため、上記累積処理及び二乗処理では、まず乗算器が使用され、その後、剰余計算器が使用される。このため、1つの処理サイクルに乗算器と剰余計算器とが交互に2回ずつ使用されることになる。つまり、乗算器と剰余計算器とは同時に使用されることがないため、これらの使用効率が悪く、全体的な演算処理の時間が長くなるという問題があった。一方、処理の高速化を図るため、上記累積処理と二乗処理とを並列化させることも考えられるが、その場合、乗算器と剰余演算器とをそれぞれ2組備えなくてはならず、回路規模が大幅に増大してしまうという問題があった。

（発明が解決しようとする課題）

このように、従来のべき乗剰余計算方式では、累積処理と二乗処理とをシリアルに実行し、しかも各処理において乗算と剰余算とは異なる時間に行われるため、全体的な演算時間が長くなってしまうという問題があった。また、上記累積処理と二乗処理とを並列化しようとすると、回路規模が大幅に増大してしまうという問題があった。

本発明は、回路規模の増大を抑えつつ、演算処理の高速化を図ることができるべき乗剰余計算方式を提供することを目的とする。

［発明の構成］

（課題を解決するための手段）

本発明は、第1、第2及び第3のレジスタ、乗算器、剰余計算器並びに制御手段を備え、与えられた整数M，N，E（$-\sum_{i=0}^{k} e_i \cdot 2^i$，$e_i$ = 1又は0）から、$C = M^E \bmod N$なる整数Cを計算するべき乗剰余計算方式において、前記制御手段が、以下のように二乗処理及び累積処理の実行を制御することを特徴としている。

即ち、制御手段は、まず初期値として前記第1のレジスタに1、前記第2のレジスタにM^2、前記第3のレジスタにMをそれぞれ設定する。

制御手段の制御により実行される二乗処理は、iを0から順に1ずつ増す度に、前記第2のレジスタの格納値と前記整数Nとを前記剰余計算器に供給しその法Nの演算結果を前記第2及び第3のレジスタに格納する第1のステップと、この第1のステップで求められた演算結果を前記乗算器に供給してその二乗値を求め前記第2のレジスタに格納する第2のステップとからなる。

また、制御手段の制御により実行される累積処理は、処理の実行を制御するとともに、iを0から順に1ずつ増す度に、e_iが1であるときのみ、前記第1のステップと同時に、前記第1のレジス

タの格納値と前記第3のレジスタの格納値とを前記乗算器に供給しその結果を前記第1のレジスタに格納するとともに、前記第2のステップと同時に、前記第1のレジスタの格納値と前記整数Nとを前記剰余計算器に供給しその法Nの演算結果を前記第1のレジスタに格納する処理である。

（作用）

本発明では、二乗処理の処理ステップが従来の処理ステップとは異なり、まず第1のステップで剰余演算が行なわれ、第2のステップで乗算が行われる。このため、第1ステップでは、乗算器が空き状態、第2ステップでは剰余計算器が空き状態となっている。従って、累積処理では、上記第1ステップで乗算器を使用し、第2ステップで剰余計算器を使用している。つまり、本発明よれば、第1及び第2のステップでは、乗算器と剰余計算器とが同時に使用される。従って、本発明では乗算器と剰余計算器の使用効率が改善され、少ないハードウェア量で二乗処理と累積処理とを並列に処理することができ、演算処理の高速化を図ることができる。

（実施例）

以下、図面に基づいて本発明の実施例について説明する。

第1図は一実施例に係るべき乗剰余演算装置の構成を示すブロック図である。

この装置は、C，R，Wの3つのレジスタ11，12，13と、本装置の主体をなす乗算器14及び剰余計算器15と、5つのセレクタ16～20と、これらを制御する制御部21とにより構成されている。Cレジスタ11，Rレジスタ12及びWレジスタ13は、処理途中の結果を一時的に格納するレジスタで、最終的な暗号文はCレジスタ11内に格納される。Cレジスタ11とRレジスタ12は例えば1024ビット、Wレジスタ13は例えば512ビットのレジスタとなっている。乗算器14は、2数の積を求める演算手段で例えば512ビット×512ビットの演算能力を持つ。剰余計算器15は、例えば1024ビットの入力に対し、与えられた法Nの下で、剰余計算を行う演算手段である。乗算器14への入力は、セレクタ16，17でCレジスタ11，Rレジスタ12及びWレジスタ13の中から選択され、乗算結果はセレクタ18によって選択されたCレジスタ11及びRレジスタ12の中のいずれか一つに格納される。剰余計算器15への入力は、セレクタ19でCレジスタ11及びRレジスタ12のいずれか一方から選択される。剰余算結果は、Rレジスタ12及びWレジスタ13の両方若しくはCレジスタ11にセレクタ20によって選択されて格納される。制御部21は、後述する処理手順に従って各セレクタ16～20の選択信号を出力し、データの流れを制御する。

第2図に上記制御部21の処理手順を示す。始めに512ビット程度の整数であるNと、N－1以下の整数であるM，Eとが与えられる。法の数Nは、制御部21から剰余演算器15に受け渡される。まず、初期値としてCレジスタ11に1、Wレジスタ13にMがそれぞれ格納される。制御部21内のループ変数iは、0にリセットされる（21）。以下、Cレジスタ11の内容をC、Rレジスタ12の内容をR、Wレジスタ13の内容をWとして記述する。先ず、W＊Wの乗算結果がRレジスタ12に格納される（22）。

次に、べき乗数E（$=\Sigma e_i \cdot 2^i$）を、

$e_k \cdot e_{k-1} \cdot \cdots \cdot e_1 \cdot e_0$

と2進数表現したときの最下位ビットe_0が0であれば（23）、二乗処理のみを実行する（24）。この二乗処理は、剰余計算器15を用いてR mod Nを求め、その結果をRレジスタ12とWレジスタ13の両方に格納する第1のステップS1と、乗算器14を用いてR＊Rを求め、その結果をRレジスタ12に格納する第2のステップS2とからなる。e_0が0の場合の処理はこれで終了する。

一方、e_0が1の場合には（23）、上記と同じ二乗処理（25）と同時に累積処理（26）を実行する。この累積処理（26）は、上記第1のステップS1と同時に、乗算器14を用いてC＊Wを求め、その結果をCレジスタ11に格納

するステップＳ１′と、その後、上記第２のステップと同時に、剰余計算器１５を用いてＣ mod Ｎを求め、その結果をＣレジスタ１１に格納するステップＳ２′とからなる。つまり、上記第１のステップと第２のステップでは、乗算器１４と剰余計算器１５とが同時に使用されている。これら乗算と剰余計算の両方が終了し、その結果の各レジスタへの格納をもってe_0が１の場合の処理は終わる。

次に制御部２１は、ループ変数ｉをインクリメントし（２７）、同様の処理を繰返す。ｉがｋ＋１になったら処理は終了する（２８）。そのときのＣレジスタ１１の内容がM^E mod Ｎの結果である。

第３図は、Ｅ＝１１＝$(1011)_2$の場合に、各時点で各レジスタが保持している内容を示す図である。この図に示すように、各ループでの処理は２つのステップで終了する。そして、Ｃレジスタ１１の格納値は、$e_i=1$のときに、Ｃ＊Ｗを先に実行し、Ｃ mod Ｎを後に実行することにより得ているので、最終的には、M^{11} mod Ｎが求められる。Ｒレジスタ１２の格納値を求める際は、剰余演算が先行し、その後剰余演算結果を二乗しているが、次のループの第１ステップＳ１で再度剰余演算をすることにより、更に次のループの第１ステップＳ１′で値Ｃに乗算される正しいＷの値が求められている。

上記装置によれば、$e_i=1$のときに、乗算器１４と剰余計算器１５の同時使用が可能であるので、演算装置の使用効率が向上し、処理時間が短縮される。処理時間の短縮は、乗算器の処理速度と剰余計算器の処理速度とが等しい場合に最も顕著となり、更にべき指数Ｅを２進数で表わしたとき、１が立つビット数が多い程、上記累積処理と二乗処理の並列処理効果が奏され、Ｅのビットが全て１である場合には、従来の半分に短縮される。

【発明の効果】

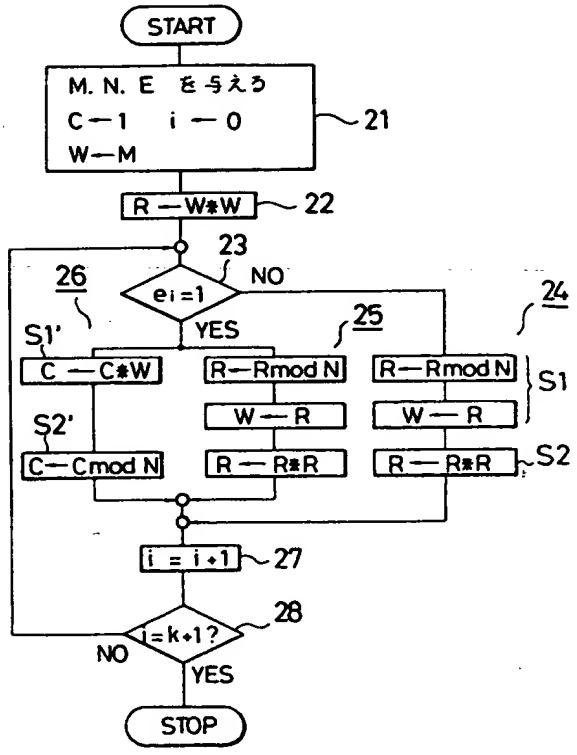
以上のように、本発明によれば、二乗処理の演算順序を累積処理の演算順序と換えることにより、乗算器と剰余演算器を同時に異なる処理に使用して累積処理と二乗処理とを並列的に行っているので、簡単な構成で処理の高速化を図れるという効果を奏する。

４．図面の簡単な説明

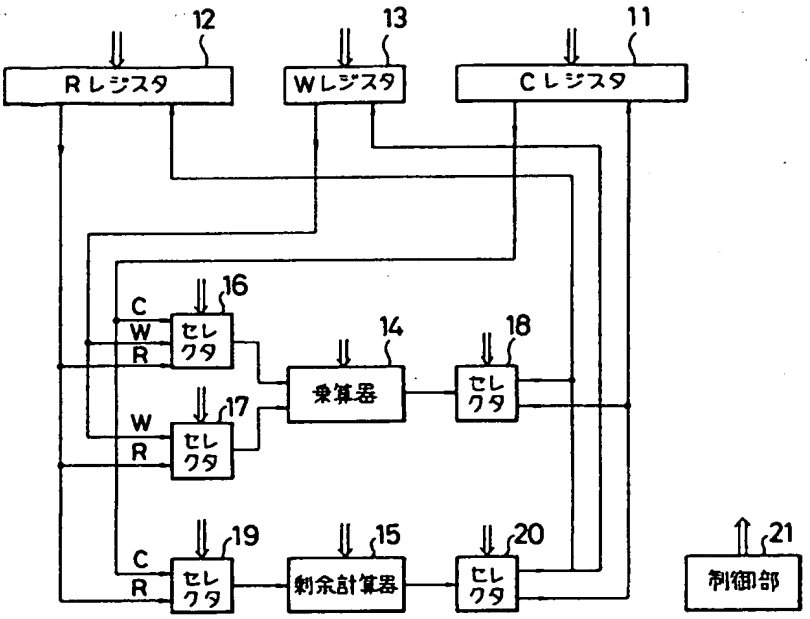
第１図は本発明の一実施例に係るべき乗剰余計算装置のブロック図、第２図は同装置の動作を説明するための流れ図、第３図は同装置を用いて計算を行った時の各時点でのレジスタの内容の一例を示す図、第４図は従来のべき乗剰余計算装置の動作を示す流れ図である。

１１…Ｃレジスタ、１２…Ｒレジスタ、１３…Ｗレジスタ、１４…乗算器、１５…剰余計算器、１６～２０…セレクタ、２１…制御部。

出願人代理人　弁理士　鈴江武彦



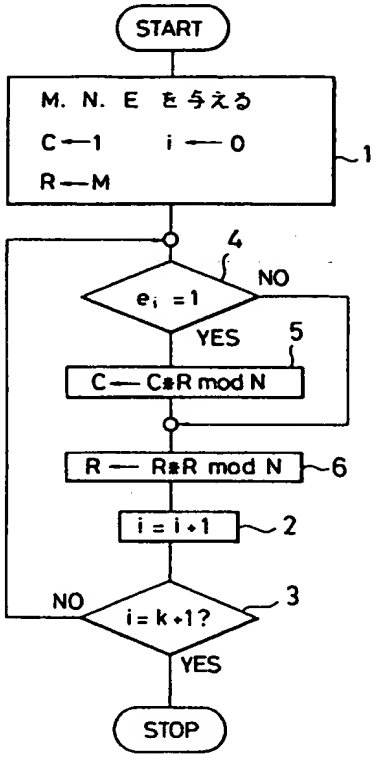
第　２　図



第 1 図

		C	R	W
i=0 (e_0=1)	初期値	1	M^2	M
	S1,S1'	M	M^2 mod N	M^2 mod N
	S2,S2'	M mod N	$(M^2 \bmod N)^2$	↑
i=1 (e_1=1)	S1,S1'	M mod N·M^2 mod N	M^4 mod N	M^4 mod N
	S2,S2'	M^3 mod N	$(M^4 \bmod N)^2$	↑
i=2 (e_2=0)	S1,S1'	↑	M^8 mod N	M^8 mod N
	S2,S2'	↑	$(M^8 \bmod N)^2$	↑
i=3 (e_3=1)	S1,S1'	M^2 mod N·M^8 mod N	M^{16} mod N	M^{16} mod N
	S2,S2	M^{11} mod N	$(M^{11} \bmod N)^2$	↑

第 3 図



第 4 図